VIRTUAL ENVIRONMENTS FOR DESIGN AND MANUFACTURE[1]

Sankar Jayaram[2], Scott Angster[3], and Kevin Lyons[4]

1 INTRODUCTION

The technology of computer aided design and computer aided manufacturing (CAD/CAM) has progressed significantly from the two-dimensional wireframe drafting systems of the 1970s to the parametric and feature-based solid modelers of the 1990s. There are two new technologies which seem to be pushing CAD/CAM into the next generation. These are virtual reality and internet communications. Virtual reality is a relatively new technology which can be regarded as a natural extension to three-dimensional graphics with advanced input and output devices. In very simple terms, virtual reality (VR) can be defined as a synthetic or virtual environment that gives a person the illusion of physical presence. "The exposure most people have to the concept of virtual reality is through reports in the media, through science magazines, and through science fiction. However, to the researchers involved in the actual science of virtual reality, the applications are much more mundane, and the problems are much more real" (Jayaram, 1996). A good discussion of virtual reality is presented by Machover and Tice (Machover, 1994), and Ellis (Ellis, 1994).

The modern product development process calls for rapid design through manufacturing cycles, agile manufacturing systems, adapting designs to suit rapidly changing customer requirements and preferences, use of centralized advanced manufacturing facilities, and outsourcing fabrication. The current suite of product development and rapid prototyping tools are not geared for these new scenarios. Virtual prototyping is a new concept which allows designers to create digital prototypes and evaluate the products thoroughly before a physical prototype is created at a remote location or by a sub-contractor. This significantly reduces time to market and increases the competitiveness of a company. However, the software tools used for virtual prototyping (from traditional CAD systems to virtual reality based manufacturing simulations) are so varied in their architecture that it is almost impossible to create a cohesive set of virtual prototyping tools for use by any organization. This is compounded by the fact that customers and sub-contractors have completely different sets of tools for performing the same tasks.

Many of the issues in virtual prototyping are addressed by virtual reality technology. The ability to view full-scale, three-dimensional models can increase the designer's productivity by allowing a more natural analysis of a model. Virtual fly-throughs of manufacturing simulation can provide a better understanding of the layout of a manufacturing plant. Virtual manufacturing can be used to test and modify numerical control codes off-line and to train workers. Virtual assembly will allow engineers to study ease of assembly and ease of handling. Virtual disassembly can be used to study repair and recycling issues. However, most current systems that have been built to exploit the power of virtual reality are limited in their expandability, customization or usability with current design software systems.

Information that is created and maintained within VR systems must be sharable and capable of being applied and utilized by complementary systems such as Computer-Aided Engineering (CAE) applications. In the case of assembly planning, this tight integration with other design and engineering systems (e.g., CAD and VR functionality with supporting input and display devices, and data exchange) will enable manufacturing engineers to evaluate, determine and select more optimal component sequencing, generate assembly/disassembly process plans, make better decisions on assembly methods (i.e., automated or manual assembly), and visualize the results.

[1] This paper represents Chapter 8 in "Concurrent Engineering: The Agenda for Success," Edited by Sa'ad Medhat, Research Studies Press Ltd., Taunton, Somerset, England, 1997.
[2] Sankar Jayaram, Washington State University, Pullman, WA, 99163-2920, jayaram@mme.wsu.edu
[3] Scott Angster, Washington State University, Pullman, WA, 99163-2920, angster@mme.wsu.edu
[4] Kevin Lyons, NIST, Gaithersburg, MD, 20899, klyons@cme.nist.gov

This chapter describes the creation of virtual environments for design and manufacture. The object-oriented architecture of such systems is described using two specific instances as examples. These two systems are VEDAM (Virtual Environment for Design and Manufacturing) and VADE (Virtual Assembly Design Environment). Preliminary results of the use of these systems is also presented. VEDAM is a very general framework for virtual reality applications in design and manufacturing, whereas VADE is specifically designed for assembly planning.

2 OTHER RELATED WORK

An overview of the application of virtual reality to CAD/CAM has been presented by Jayaram (Jayaram, 1996a). Several groups have developed systems utilizing virtual reality techniques for early design decisions through the use of virtual fly-throughs, virtual design, virtual assembly and manufacturing simulation. Washington State University has developed a system for the early design evaluation of automobile interiors. This system utilizes Pro/ENGINEER™ models that are brought directly into a virtual design environment. This work was a continuation of a feasibility study that was conducted that provided successful results in the use of virtual reality for design (Angster, 1994, 1996b).

Through joint work at the University of Illinois, Chicago, and Purdue University, a prototype, virtual reality based, computer-aided design system has been designed and implemented. The focus of this work is to allow a simplified method of designing complex mechanical parts through the use of virtual reality techniques (Trika, 1997). Work at the Georgia Institute of Technology is focusing on early design changes based on disassembly and servicing criteria (Rosen, 1995). The University of Bath in Bath, UK has developed an interactive virtual manufacturing environment. This environment models a shop floor containing a three-axis numerical control milling machine and a five-axis robot for painting. The user can mount a workpiece on the milling machine, choose a tool and perform direct machining operations, (such as axial movements or predefined sequences,) or load a part-program from memory (Bayliss, 1994). A virtual workshop for mechanical design was developed at Massachusetts Institute of Technology (Barrus, 1993). The goal of the project was to develop a simulated workshop for designers to do conceptual design work while having to take into account manufacturing processes. The National Institute of Standards and Technology has developed a Virtual Reality Modeling Language (VRML) interface for a system called VIM, or Visual Interface to Manufacturing (Ressler, 1997). This system provides visual access, using VRML, to a database containing manufacturing data, such as three-dimensional models of parts, assemblies and shop floor assembly workstations.

Deneb Robotics has available commercial software for manufacturing simulation, virtual milling, virtual spray painting, virtual arc welding and telerobotics. Most of the these systems are precompiled software tools (Deneb). Technomatix Technologies has developed several products in the area of virtual manufacturing. Robcad™ has been developed as a computer-aided production engineering tool for the analysis of robotic applications in a virtual environment. These include welding, laser cutting and painting. Robcad also allows for off-line robot programming and an open system architecture for developing user-specific features and applications. Another product, Part, has been developed for computer-aided, numerical control, process planning and programming. Part™ provides machine tools, machine setup, machining methods and cutting tools. Part will automatically create numerical control programs and process plans given computer-aided design models (Technomatix). Prosolvia Clarus AB manufacturing has created a virtual manufacturing toolkit that is adapted for Technomatix Robcad modules. Users can visualize a manufacturing simulation interactively through the use of virtual walk-throughs of detailed computer-aided design data. This toolkit features an open architecture for users to create their own applications. Clarus is also researching and developing virtual assembly tools (Proslavia).

Resolution Technologies has created Virtual Mockup™, a system for fly-through analysis of computer-aided design models. Virtual Mockup provides a set of tools that allows the user to view the models in real-time, query the models data set, remove objects from the model and save viewing motion paths for playback (Resolution).

Angster (Angster, 1993), Angster and Jayaram (Angster, 1995), and Narayanan (Narayanan, 1993), created a device-independent, object-oriented, knowledge-based system framework, called the Expert Consultation Environment. This framework gives programmers of computer-aided design applications the ability to add knowledge-based systems to their applications without having to worry about the details of knowledge-based systems programming. Jayaram, et. al., created a prototype of an object-oriented framework to support the integration of multiple software systems used in virtual prototyping of mechanical components (Jayaram, 1996). The goals of this work included the coupling of software systems which have different data requirements and the demonstration of the system in an industrial setting for the development of electronic spray-cooling components. Jayaram (Jayaram, 1989, 1990) designed an architecture which supports the creation of device-independent, customized product development tools. This architecture included design, user interface, knowledge-based assistant, and virtual manufacturing environments.Woyak, et. al. (Woyak, 1995), describe an architecture, called the Dynamic Integration System, which addresses the issues of software integration for engineering applications. This architecture is based on the concept of dynamic variables and dependency hierarchies.

Schroeder et al. (Schroeder, 1994a) of GE Corporate Research & Development, discuss a proposed procedure for designing for maintainability. Along with the accessibility of parts and fasteners, the issues of part path and swept volumes are also addressed. Swept surfaces and volumes are generated by a soleid model as it moves through time and space on an arbitrary, time-dependent trajectory. This concept has been applied to the problem of maintainability of jet aircraft engines and "safe" path planning in robot applications (Schroeder, 1994b) According to Kijima and Hirose (Kijima, 1996), "the manual handling of objects if the basic problems which still exists in the further development (of) virtual reality technology." In this paper, the authors describe the generation of object behavior and application of different models of an object in a VR environment. As stated by Fernando et al. (Fernando, 1994), "a common weakness of the existing virtual environments is the lack of efficient geometric constraint management facilities such as run-time constraint detection and the maintenance of constraint consistencies during 3D manipulations."

3 OVERVIEW OF VEDAM

As stated earlier, there are several virtual reality techniques that can assist in the design and manufacturing planning of a product. These include virtual design, virtual assembly, virtual manufacturing, and human-integrated design. A valuable design system for engineers is one that will support all of these techniques, yet be compatible with current parametric CAD/CAM systems. The architecture behind such a system should allow for the expansion and customization of the virtual environments to suit the engineer's needs. An overview of a proposed system, VEDAM, is shown in Figure 1. This figure shows the VEDAM system and its components, the Machine Modeling Environment (MME), the Virtual Design Environment (VDE), the Virtual Assembly Environment (VAE), and the Virtual Manufacturing Environment (VME). The MME is an environment that allows the user to create models of the actual machines found in the factory that will be used to produce the product. The VDE is a immersive, virtual reality based design environment that allows the user to view, scale and modify a parametric model designed in a CAD system. The VAE is an immersive, virtual environment that allows the user to analyze the assembly of parts through the direct manipulation of the parts by the user (Connacher, 1995, 1996, 1997). The VME is an immersive virtual environment that allows the user to analyze and develop process plans using a virtual factory that that replicates and functions of the real factory.

VEDAM would interface with the parametric CAD/CAM system through the main interface. During a design session, the user would have the option of entering into one of the environments via the main graphical user interface to test designs or manufacturing ideas. All required data from the CAD/CAM system would then be passed into the virtual environments. Upon exiting the virtual environments, the user would have the option of passing data back into the CAD/CAM system. VEDAM, combined with a parametric CAD/CAM system, would provide a complete system for engineers to evaluate potential designs and process plans.

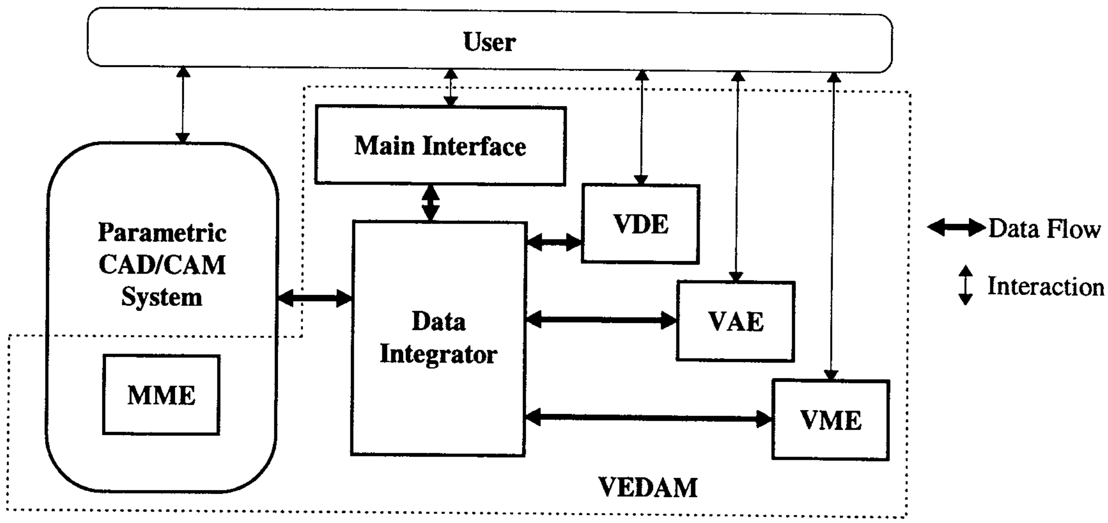

Figure 1. VEDAM System (Angster, 1996a, 1996c)

4 OBJECT-ORIENTED ANALYSIS OF VEDAM

Based upon the initial concept discussed above, object-oriented methods were used to create a prototype framework of the VEDAM system. Object-oriented methods were used due to the complexity of the system to be designed. Traditional algorithmic decomposition of the VEDAM system would produce a very unmanageable and inflexible system. By breaking the overall system down into individual classes that communicate with each other, each class can be separately created, tested and added to the existing classes. This allow for very manageable and expandable code.

The initial object-oriented analysis of the system required a description of the functional usage of the system by a user. The description was analyzed to create a set of classes needed for the VEDAM system.. These can be seen in Figures 2 through 5. Each figure shows a key group or key set of interactions of classes. Each link between classes represents a dependency between those classes. An open circle represents a "use" relationship, whereas a filled circle represents a "has" relationship. Figure 2 shows those classes that are mostly related to the user of the system. Figure 3 shows those classes related to the machining operations. Figure 4 shows those classes that are related to the assembly process and Figure 5 shows how each class that has a visual representation in the VR environment uses the geometry class. The main classes are the *model manager* class, the *interaction manager* class, the *input manager* class, *output manager* class, the *graphical user interface* class, the *hardware* class, the *human model* class, the *user* class, the *machine* class, *fixture* class, the *cutting tool* class, the *stock* class, the *workpiece* class, the *assembly* class, the *constraint* class, the *parameter* class, the *switch* class, the *controller* class, the *factory* class, the *process plan* class, the *cutting properties* class and the *geometry* class. A brief description of each class follows.

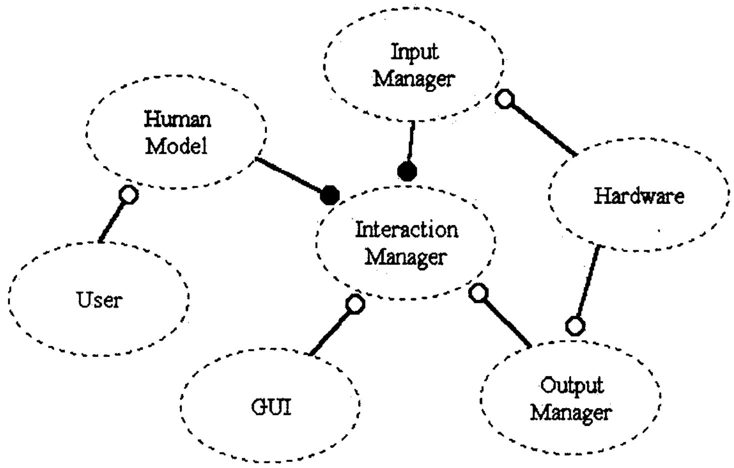

Figure 2. User Related Classes

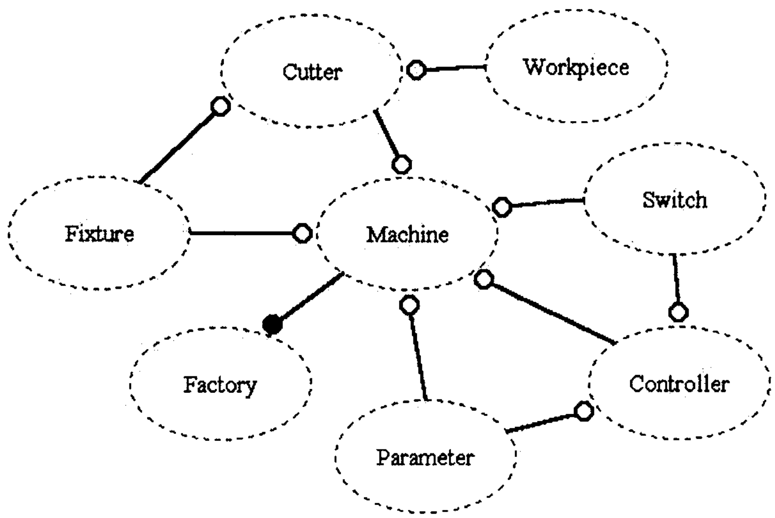

Figure 3. Machining Related Classes

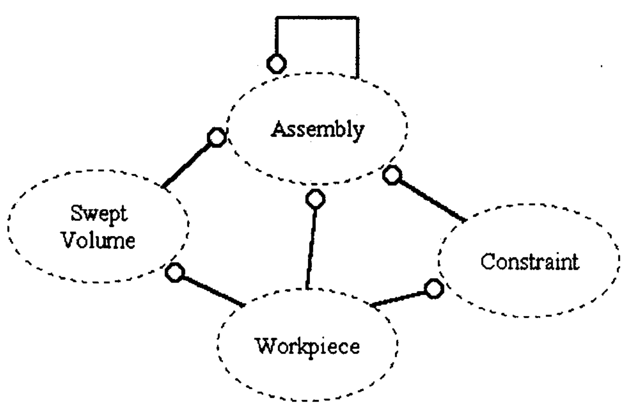

Figure 4. Assembly Related Classes

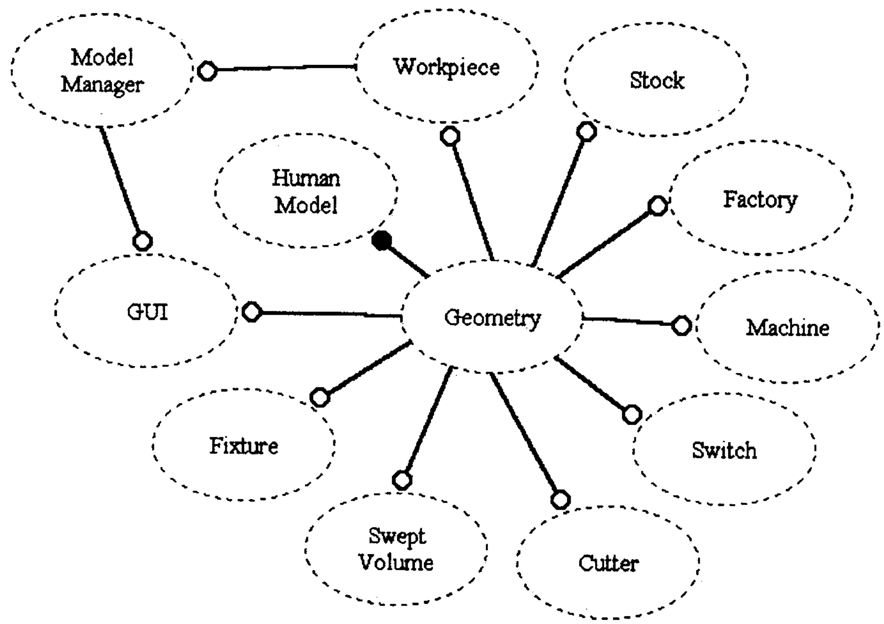

Figure 5. Geometry Related Classes

Interact Manager Class: The *interact manager* class is the center of the VEDAM system. This class is responsible for all interaction that occurs during the use of the VEDAM system between the user of the system and the virtual environment.

Model Manager Class: The main responsibility of the *model manager* class is to act as a data integrator between the virtual reality system and traditional CAD/CAM systems.

Geometry Class: The *geometry* class is used by many other classes to create a geometric model for visual representation within the virtual environments.

Hardware Class: The *hardware* class is a virtual class for all input and output hardware that will be used by the VEDAM system.

Input Manager Class: The *input manager* class is responsible for getting all the information from the classes derived from the hardware class that represent an input device.

Output Manager Class: Similar to the *input manager* class, the *output manager* class is responsible for passing any information to the classes derived from the *hardware* class that represent an output device.

Graphical User Interface Class: There are two instances of the *graphical user interface* class. One will present the menu system of the main menu to the user at the startup of the VEDAM system. The other responsibility of this class will be the presentation of the immersive menu system to the user.

User class: The *user* class is used to store the information that will be required by the *human model* class.

Human Model Class: The *human model* class is responsible for providing an accurate representation of the user within the virtual environments.

Factory Class: The *factory* class will be used to store the information about the virtual factory.

Machine Class: The *machine* class is used to represent a machine in the actual factory.

Switch Class: The *switch* class is responsible for storing information concerning the functionality of a switch, button or lever that is located on a specific machine.

Controller Class: The *controller* class is used by the *machine* class to represent the controller on the machine.

Fixture Class: The *fixture* class represents a fixture used on a machine.

Cutter Class: The *cutter* class is used to represent a cutter that is attached to a machine.

Workpiece Class: The *workpiece* class represents the current state of the part that is being analyzed in the virtual manufacturing environment.

Stock Class: The *stock* class is used to represent the piece of stock that a workpiece is being created from.

Assembly Class: The *assembly* class represents an assembly of parts that was created in Pro/Engineer.

Swept Volume Class: The *swept volume* class is used to store the volume created by the sweeping motion of parts as they are assembled.

Parameter Class: The *parameter* class is used by many classes for storing parameters and their associating properties. This would include minimums, maximums, current values and methods for changing the values.

Process Plan Class: The *process plan* class is used to store the overall process plan that is created using the virtual manufacturing environment and the virtual assembly environment.

Cutting Properties Class: As the machining process is going, the various cutting properties are calculated and stored in instances of this class. This class would store such information as cutting time, forces, surface quality and power consumption. All of these properties would be saved as functions of time.

5 INITIAL IMPLEMENTATION AND RESULTS

After completing the object-oriented analysis of the VEDAM system, an object-oriented design produced an initial implementation of the system. The initial implementation provides the user with a virtual manufacturing environment and a virtual design environment. A virtual assembly environment was under parallel development at the time of the VEDAM implementation and current work is being done to integrate the systems. These systems were created on a Silicon Graphics Crimson™ Workstation with Reality Engine™ Graphics. All classes were developed using C++ and the graphics were created using Performer™ 2.0. The virtual reality hardware used in this implementation include a Virtual Research VR4™ helmet, a Virtual Technologies 22-sensor Cyberglove™, and an Ascension Flock of Birds™ tracking system with an extended range transmitter and six birds.

The virtual manufacturing environment includes a table-top milling machine, a table-top lathe, and a water jet. These can be seen in Figures 6, 7, and 8 respectively. All three of these machines are numerically controlled through the use of word address format numerical control codes. A virtual controller on each system provides the functionality of an actual controller. This includes buttons for power, axial movements, setting a floating zero control, loading a part program, and running the program. A graphical user interface (GUI) provides the user a means for selecting numerical control codes. The virtual design environment is linked directly to the CAD/CAM database through the use of proprietary database interface software. All interaction with the VDE is done through the use of the GUI. The VDE allows the user to select a model to analyze, view parameters of the model, modify the values, and regenerate the model. The user will see the modified model in the virtual environment. A menu of the VDE on the GUI can be seen in Figure 9.

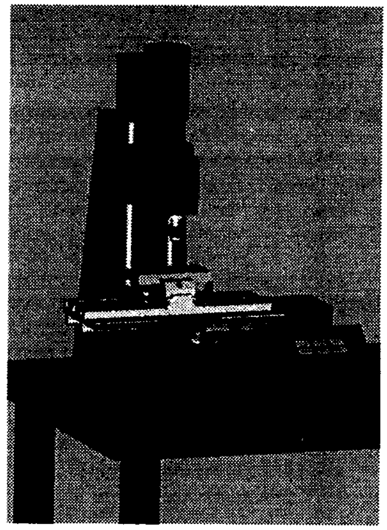

Figure 6. Table-Top Milling Machine

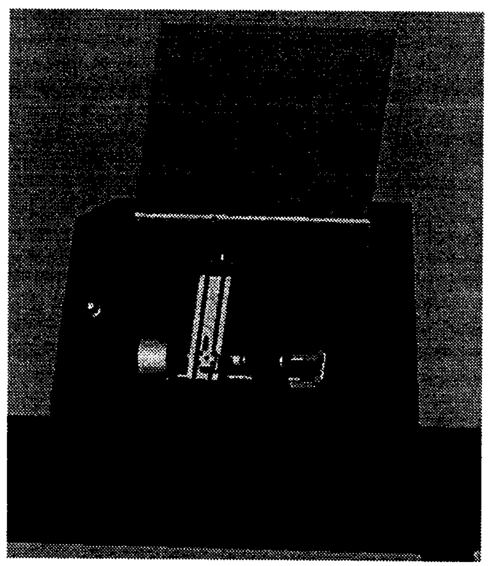

Figure 7. Table-Top Lathe

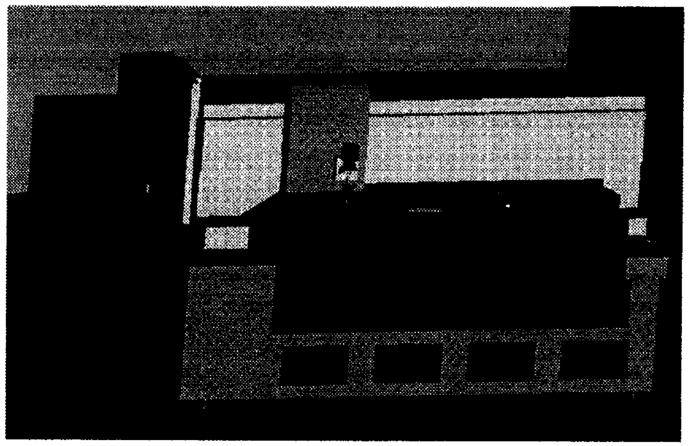

Figure 8. Water Jet

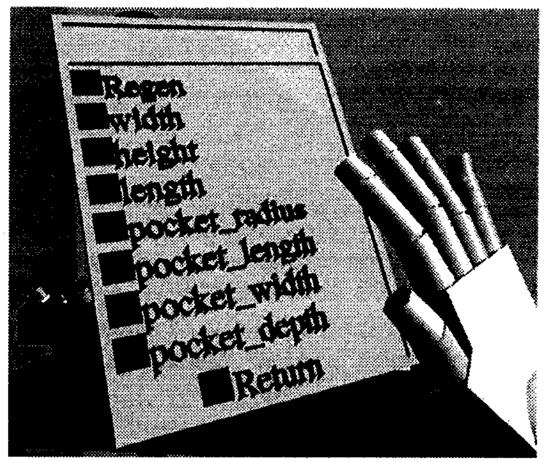

Figure 9. Graphical User Interface

This system uses most of the classes that were identified during the analysis phase. The *model manager* implemented currently supports the transfer of Pro/ENGINEER™ models from Pro/ENGINEER™ into the VEDAM system. These models include the machines, the parts and the human model. The *interaction manager* has been created to respond to keyboard entry and manages all collision detection between the human model and the environment. *The input manager* controls the input of the keyboard, the data flow of the Cyberglove™ data and the data flow of the Flock of Birds™ data. The *output manager* is responsible for the graphical output of the system to either the monitor or the helmet-mounted display. The *machine* class was created around the development of the table top milling machine. However, generic methods were written that would provide interactive capabilities for any machine developed in the machine modeling environment. A *graphical user interface* class provides the three-dimensional interactive menu system for the user.

Test cases have been run on all three machines to test the virtual manufacturing environment for accuracy when compared with real machining. The test involved the comparison of real setup times and machining times with virtual setup and machining times of a part that was created in Pro/ENGINEER™ and Pro/Manufacture™. These times were also then compared to those machining times provided by Pro/Manufacture™ and a estimated setup time provided by a manufacturing expert. The same procedures and numerical control codes were used in both the actual and virtual environments. The test showed that the virtual manufacturing environment was able to provide a better estimate of setup times for proposed process plans than an experts estimate. Details of these tests are presented by

Angster (Angster, 1996a, 1997). The virtual design environment demonstrated the ability to view and modify models in a three-dimensional environment through the direct manipulation of a CAD system's database.

6 BACKGROUND OF VIRTUAL ASSEMBLY DESIGN ENVIRONMENT

During a typical design cycle, a product is designed, a prototype is built, changes to the design and production processes are added, and a new prototype is constructed. Once this portion of the design cycle has been completed, the product can then be produced. Often, a part or system is designed without a great deal of consideration for the environment in which it will be assembled. This can be, and often is, a time consuming and expensive method of design. In order to achieve a greater efficiency in the design cycle, ways to cut production cost and time to market need to be investigated. With the development of a virtual prototype, both the design and production costs mentioned above can be addressed. If a virtual prototype can be developed for a part or assembly and output to a virtual environment for evaluation, the production costs and time to market could be substantially reduced. The main goal of the VADE system is to provide the designer with the tools necessary to evaluate assembly considerations without the cost and time expense of the traditional design method. The main problems addressed are as follows:

1. Creation of a virtual environment.
2. Data transfer from a CAD system.
3. Use of CAD assembly information.
4. New information generated.
5. Evaluation and verification of results.

Creation of a Virtual Environment

The creation of the virtual environment involves creating an environment that is flexible from the programmer's standpoint and usable from the user's standpoint. The system needs to be especially robust because the end users of this system will have varying hardware and software setups, system requirements, and computer ability. Also to be addressed are the ways that the user interacts with the system. The data required to track the position of the head, the hands, and the finger joint angles is needed to provide the user with a believable immersive experience. Once this information is obtained, it is necessary to provide the user with an intuitive way to grab the parts and manipulate them in the virtual environment. Finally, the physical constraints for assembling the component need to be mimicked in the virtual environment.

Data Transfer from a CAD System

The problem of data transfer from a CAD system is of great importance to this research. Once the virtual environment has been created, data for the user to manipulate, both graphically and computationally, must be obtained. First, the user will need a graphical representation of the parts and assemblies he/she is to assemble and second, the user will also need the relationships between the parts and assemblies so that he/she can accurately assemble the system. The types of information to be obtained from the CAD system are:

1. The graphical representation of the part/sub-assembly.
2. The number of parts/sub-assemblies.
3. The names of parts/sub-assemblies.
4. The final locations and orientations of parts/sub-assemblies.
5. The constraint relationships between the assembled parts/sub-assemblies.

The graphical representation of the part is crucial because the image must convey the representation of the geometry well enough to be believable, but simple enough not to slow down the system. The remainder of the information to be gathered from the CAD system is required to allow the user to assemble the parts in a natural and intuitive way. The main focus of this portion of the research was to find a way to extract this information for the assembly from the CAD system and bring that information into the VADE system. This is a significant problem because the usability of the VADE system will rely on how well the user can interact and perform actual assembly operations within the environment. Critical in this interaction is the behavior of the parts in the environment with respect to each other and the user.

Use of CAD Assembly Information

The next step in the creation of the VADE system is the development of methods for using the data obtained from the CAD system. This includes importing a graphical representation of the part, checking final part locations and orientations, checking part constraints, and imposing these constraints on the part's motion. These methods should be efficient and thorough, without sacrificing model or constraint fidelity.

New Information Generated

The purpose of employing a tool similar to the VADE system is to gain some insight into the assembly process in question. Information should be returned from such a system that will allow the designer to improve on the design of the product or the processes involved in assembly. Information generated by the VADE system such as assembly sequencing, swept volume, space requirements, human factors information, etc. should be made available to the designers analyzing the current design and future designers working with subsequent future iterations of the assembly. Information of this type should be incorporated into the actual CAD database containing the assembly and the designers' intent.

Evaluation and Verification of Results

Once the user has performed the assembly in the virtual environment, it is important to verify that the results obtained correspond to the results obtained from the actual physical assembly operations. Another consideration in the design of the system is the recording and verification of the part trajectories as they travel along their paths toward final assembly location. Data management and optimization is an important issue since large amounts of data are required to give the designer an accurate description of the path of the part.

7 PROTOTYPE VADE SYSTEMS

In order to gain insight into the functioning of a complete VADE implementation, two prototype systems were created in the process of this research. These initial prototypes were built upon one another to extend the range of knowledge about the desired functionality of this type of application. Problems relating to data extraction in the first prototype were partially addressed by the second. Both initial prototypes used the same system configuration, including the same "core" for creating the virtual environment (Connacher, 1995, 1997). In the case of these prototypes, the assembly models were generated in Pro/ENGINEER™ and subsequently assembled using the constraint conditions supplied by the Pro/ENGINEER™ interface. No constraint information was implemented except checking the final locations and orientations of the part and gripping consisted of attaching the part to the fingertip of a hand model.

The main concern was the development of methods for transferring data between the solid modeling system and the assembler. Several different formats of data transfer were investigated and it was determined that, for this prototype implementation, stereolithography files would be the simplest to generate and convert. Other file formats which were considered include Render™, Inventor™, and IGES. IGES, although the international standard, is too complex for a simple situation where the main aim is to obtain the polygonal representations and relative transformations. Render and Inventor formats contained more information than was necessary for the prototype implementation. Overall, the stereolithography format provided sufficient information and was relatively simple to implement. It was determined through investigating the organization of the stereolithography files that the values required to perform the translation of the part to its final assembled position could be easily extracted from the assembly file, but the final orientation could not.

8 DATA TRANSFER FROM CAD SYSTEM

The interaction between the VADE and CAD systems is of primary importance to the designer. Automation of this link is required so that the user can efficiently and effectively use the system to its full potential. Also, the user must be able to "step out" of the CAD system and "into" the VADE system with minimal effort. The CAD system chosen to be used for this implementation of VADE was Pro/ENGINEER™.

Two methods of interaction between the CAD system and VADE were investigated. The first of these methods of interaction is to start the VADE system from within the CAD system (e.g. - a menu item, etc.). The benefit of this type of interaction is the user's direct ability to access the VADE system with a minimum of knowledge and effort. The second method requires the user to generate variations on the same design or several totally different designs within the CAD system, generate the appropriate data files, and then exit the CAD design environment. The user would then enter the VADE system to evaluate the designs. Ideally, the designers would create an assembly in the CAD system, and with the click of a mouse button, enter the virtual assembly design environment to test the assembly/design assumptions they have made in the creation of the design. Two primary disadvantages of this of interaction are: a) the complexity of the software setup, and b) the prohibitive cost of VR hardware. Hence, for the research presented in this chapter, the interaction with the CAD system was separated from the interaction with VADE.

Data Exchange

The primary information required by the VADE system is a graphical representation of the model itself. The graphical format chosen for the final implementation of the VADE system was the Inventor™ file format developed by SGI for use with the OpenInventor™ graphics library. It was determined that automated data transfer could be used to obtain the following information.

A. **number** of parts/sub-assemblies - integer value
B. **names** of parts/sub-assemblies - character strings
C. **final locations and orientations** of parts/sub-assemblies - 4 x 4 transformation matrices consisting of 16 floating point values
D. **constraint relationships** between the assembled parts/sub-assemblies.
   1. **type** of constraint - `ALIGN` or `MATE`
   2. **geometry information** for mating or aligning object
      a) **axis** - 2 point or 6 floating point values
      b) **plane** - three vectors and an origin point or twelve floating point values
      c) **offset** information about the constraint

The information is obtained through automated transfer from the CAD system using Pro/DEVELOP™, the developer's toolkit for accessing the Pro/ENGINEER™ database. This access is not possible without using this module because of the proprietary nature of the database. Using this toolkit, automated data exchange between the CAD system and VADE proved to be greatly simplified. To perform the data transfer, a new menu item was added to Pro/ENGINEER™'s graphical user interface. This menu item, Pro/VADE, starts the data transfer process using Pro/DEVELOP™'s functionality. For this research, it was decided that only the first level within the assembly tree would be transferred to VADE. Although complex designs are likely to contain several sub-assemblies, any sub-assembly that is part of an assembly can be assembled using the VADE system in a separate session, providing a great simplification to the requirements of the system.

The information first obtained from the CAD system is the number of level-one parts. Next the names of these parts and their model IDs are obtained. Then, the final locations and orientations of each component in the assembly are extracted from the Pro/ENGINEER™ data base. The information is stored as a 4 x 3 transformation matrix. The final type of information needed from the Pro/ENGINEER™ data base is the constraint information for the assembly operations. An assembly test case was performed to determine exactly what kind of constraint information was needed to perform the assembly in reality and in Pro/ENGINEER™ (Connacher, 1996a). The information obtained from the CAD system about axial and planar constraints is as follows. For an axis, the two-points in space defining the ends of the graphical line representing the axis are obtained. For a planar constraint, three unit vectors and an origin are obtained, thus defining a plane of constraint. The data is then compiled into a VADE data file.

9 PROTOTYPE IMPLEMENTATION AND RESULTS

The first task accomplished in implementing VADE was the object-oriented analysis and design of the system (Connacher, 1996a). The virtual environment, including stereo viewing and head tracking, was then implemented. The "assembly station" consists of a desk where the assembly operations are performed. This desk was created by taking measurements from a physical desk and generating a crude facsimile using Pro/ENGINEER™. There are also "bins" located in shelving above the desk, so the user can easily select the part for assembly. A ceiling, floor, and walls (one with a window) were added to give the designer a better sense of realism. Once creation of the environment was complete, the graphical representations of the assembly and its parts were imported into the system. Figure 10 shows the assembly environment as viewed by the user.

To allow the user to manipulate the objects within the environment, gripping and releasing of objects was implemented next. To have an intuitive interface between the user and the VADE system, it was desirable to simulate the human hand realistically within the virtual environment. This was accomplished by employing a Virtual Technologies CyberGlove™ to measure the bending and abduction of the fingers. To simulate, or abstract, a "skin" on the finger, a series of line segments (sensors) were attached each finger of the hand (Connacher, 1996a).

This still did not allow the user to perform the assembly operations necessary to complete the assembly, so the constraints and constraining the motion of parts were then created. Along with the constraints on part assembly, a tolerance for final part placement is needed to compensate for the inherent "inaccuracies" of the hardware employed. Unless there is some type of tolerance, exact alignment of axes and planes is impossible. The final step in the development of the system is the recording of the trajectory information of the path the part travels through space to its final location and orientation. This provides the method for verifying the accuracy of the system as well as giving the user the ability to possibly "reserve" space for the assembly process for future iterations of the design.

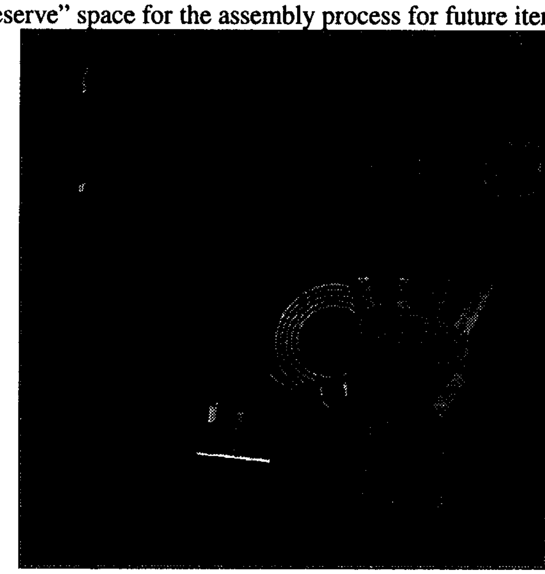

Figure 10. Virtual Environment of the VADE System

10 CONCLUSIONS

This chapter has described a framework of a virtual reality based product development system that would aid engineers in the conceptual design and manufacturing process planning stages of a product. Through the use of object-oriented methods, this framework provides an open architecture that allows the easy customization and expansion by the user. The open architecture provides the flexibility to integrate this system with any parametric CAD/CAM system and use any number of input and output devices. The initial implementation demonstrated this feature. As new classes were developed, they were inserted into the system, and only minor changes to the existing classes were made. Also, as more sophisticated methods were developed, only the class that the method belonged to needed to be modified, without affecting the performance of the rest of the classes. These two features allowed a step-wise development of the implementation.

By linking this framework to an existing parametric CAD/CAM system, engineers can immediately obtain the benefits of using virtual prototyping techniques. The analysis of designs in a true, three-dimensional environment, manufacturing the part on replications of the actual factory machines, and the assembly of mating parts are all valuable tools for product development. The initial implementation of this system has demonstrated the feasibility and usefulness of such systems.

The design and implementation of the VADE system was also successful on many points. The prototype VADE system offers designers a unique tool for achieving the goals of a useable, manufacturable, and assemblable product. This system will allow the user to complete the typical design cycle by using the traditional approach of designing an assembly in a CAD system, creating a virtual prototype, making changes to the design based on studies of this prototype, and creating a new and better design based on the information gathered from this cycle.

It is hoped that this work will provide a basis for future work to be performed in the areas of virtual manufacturing and virtual assembly. The success of the VADE system provides a starting point for future implementations of similar systems which will include a more flexible environment, enhanced gripping capability, physics-based modeling, extended constraint functionality, and more complete softzone generation capability.

11 ACKNOWLEDGMENTS

Continued research in this area is being funded by the National Institute of Standards and Technology (N.I.S.T.), Manufacturing Systems Integration Division.

The authors would like to thank Light Machines Corporation for giving permission to model their machines, the proLight Machining Center and the spectraLight Turning Center, for the virtual manufacturing work.

12 DISCLAIMER

Certain commercial equipment, instruments or materials are identified in this chapter. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products identified are necessarily the best available for the purpose.

13 REFERENCES

Angster, S.R. (1993), *An Object-Oriented, Knowledge-Based Approach to Multidisciplinary Parametric Design*, M.S. Thesis, Virginia Polytechnic Institute and State University, December 1993.

Angster, S.R., Gowda, S., Jayaram, S. (1994), *Feasibility Study on Virtual Reality for Ergonomic Design*, Presented at the IFIP 5.0 Workshop on Virtual Prototyping, Providence, Rhode Island, USA, September 1994.

Angster, S.R., and Jayaram, S. (1995), *An Object-Oriented, Knowledge-Based Approach to Multidisciplinary Parametric Design*, AIAA 33rd Aerospace Sciences Meeting, Reno, Nevada, January 1995, AIAA 95-0242.

Angster, S.R. (1996a), *VEDAM: Virtual Environments for Design and Manufacturing,* Ph.D. Dissertation, Washington State University, December 1996.

Angster, S.R., Gowda, S., Jayaram, S. (1996b), Using VR for Design and Manufacturing Applications - A Feasibility Study, 1996 ASME Design Engineering Conferences and Computers in Engineering Conference, Irvine, August 1996.

Angster, S.R. and Jayaram, S. (1996c), *VEDAM: Virtual Environments for Design and Manufacturing*, Symposium on Virtual Reality in Manufacturing Research and Education, Chicago, October 1996.

Angster, S.R., Jayaram, S., Hutton, D., (1997), *Case Studies on the Use of Virtual Reality for an Integrated Design and Manufacturing System*, 1997 1996 ASME Design Engineering Conferences and Computers in Engineering Conference, Sacramento, September 1997.

Barrus, J.W. (1993), *The Virtual Workshop: A Simulated Environment for Mechanical Design*, Ph.D. Dissertation, Massachusetts Institute of Technology, September 1993.

Bayliss, G.M., Bower, A., Taylor, R.I., and Willis, P.J. (1994), *Virtual Manufacturing*, Presented at CSG 94 - Set Theoretic Modelling Techniques and Applications, Winchester, UK, April 13-14, 1994.

Bennet, G.R. (1995), *Virtual Reality Simulation Bridges the Gap Between Manufacturing and Design*, Mechanical Incorporated Engineer, April/May 1995.

Connacher, H., Jayaram, S., and Lyons, K. (1995), *Virtual Assembly Design Environment*, Proceedings of the 15th ASME International Computers in Engineering Conference, Boston, MA, September 17-21, 1995.

Connacher, H. I. (1996), *A Virtual Assembly Design Environment*, M.S. Thesis, Washington State University, Pullman, WA., May 1996.

Connacher, H., Jayaram, S., and Lyons, K. (1997), *Virtual Assembly Using Virtual Reality Techniques,* accepted for publication in Computer-Aided Design, 1997.

Ellis, S.R., (1994), *What are Virtual Environments?*, IEEE Computer Graphics and Applications, Vol. 14, No. 1., pp. 17-22, 1994.

Fernando, T., Fa, M., Dew, P., and Munlin, M. (1994), *Constraint-based 3D manipulation techniques within Virtual Environments*, Virtual Reality and Applications, edited by R. Eranshaw et. al., Academic Press Ltd., London, 1994.

Jaques, M., Strickland, P., Oliver, T.J. (1995), *Design by Manufacturing Simulation: Concurrent Engineering Meets Virtual Reality*, Mechatronics, 1995.

Jayaram, S. (1989), *CADMADE - An Approach Towards a Device-Independent Standard for CAD/CAM Software Development*, Ph.D. Dissertation, Virginia Polytechnic Institute and State University, April 1989.

Jayaram, S., and Myklebust, A. (1990), *Towards a Standardized Environment for the Creation of Design and Manufacturing Software*, proceedings of the International Conference on Engineering Design, Dubrovnik, Yugoslavia, August 1990.

Jayaram, S. (1996a), *Virtual Reality and CAD/CAM*, The CAD/CAM Handbook, edited by Machover, C., The McGraw-Hill Company, p. 410, 1996.

Jayaram, U., Bakkom, J., Jayaram, S., and Seaney, K. (1996b), *Integration Framework for Virtual Prototyping Software Tools*, Final Report, Phase 1, Department of Commerce SBIR Grant, February 1996.

Kijima, R. and Hirose, M. (1996), *Representative Spherical Plane Method and Composition of Object Manipulation Methods*, Proceedings, IEEE Virtual Reality Annual International Symposium, Raleigh, NC, 1996.

Machover, C. and Tice, S.E. (1994), *Virtual Reality*, IEEE Computer Graphics and Applications, Vol. 14, No. 1. pp. 15-17.

Narayanan, P. (1993), *An Object-Oriented Framework for the Creation of Customized Expert Systems for CAD*, M.S. Thesis, Virginia Polytechnic Institute and State University, May 1993.

Rosen, D.W., Bras, B., Mistree, F., and Goel, A. (1995), *Virtual Prototyping for Product Demanufacture and Service Using a Virtual Design Studio Approach*, ASME Computers in Engineering Conference, Boston, MA, 1995.

Ressler, S., Wang, Q., Bodardy, S., Sheppard, C., and Seidman, G., *Using VRML to Access Manufacturing Data*, Proceedings of VRML 1997: Second Syposium of the Virtual Reality Modeling Language, Monterey, CA, February 1997.

Schroeder, W.A., Langan, J.A., and Linthicum, S. (1994a), *Design for Maintainability*, Internal Report, General Electric Corporate Research and Development, Schenectady, NY, URL: "http://www.ge.com/crd/img_and_vis_lab.html".

Schroeder, W.A., Lorensen, W.E. and Linthicum, S. (1994b), *Implicit Modeling of Swept Surfaces and Volumes*, Internal Report, General Electric Corporate Research and Development, Schenectady, NY.

Trika, S.N., Banerjee, P., and Kashyap, R.L. (1996), *Virtual Reality Interfaces for Feature-Based Computer-Aided Design Systems*, accepted for publication in Computer-Aided Design, 1996.

Woyak, S., Malone, B., Myklebust, A. (1995), *An Architecture for Creating Engineering Applications: The Dynamic Integration System*, Proceedings of the Computers in Engineering Conference and the Engineering Database Symposium, Boston, MA, September 1995.

Prosolvia Clarus AB, *http://www.clarus.se/.*

Deneb Robotics, *http://www.deneb.com/.*

Resolution Technologies, *http://www.restec.com/.*

Technomatix Technologies, LTD., *http://www.technomatix.com/.*